

⑫ 公表特許公報(A)

平5-509184

⑬ 公表 平成5年(1993)12月16日

⑭ Int. Cl.⁹ 識別記号 庁内整理番号 審査請求 未請求
 G 06 F 15/80 3 9 0 H 9190-5L 予備審査請求 有 部門(区分) 6(3)
 12/08 T 7608-5B
 15/16 9190-5L

(全 9 頁)

⑯ 発明の名称 並列プロセッサアレイのための仮想処理アドレス・命令ジェネレーター

⑰ 特 願 平3-510619

⑱ 翻訳文提出日 平4(1992)11月30日

⑲ 出 願 平3(1991)5月14日

⑳ 国際出願 PCT/US91/03345

㉑ 国際公開番号 WO91/19268

㉒ 国際公開日 平3(1991)12月12日

優先権主張 ㉓ 1990年5月29日 ㉔ 米国(US) ㉕ 529,947

⑳ 発 明 者 ジャクソン、ジェームス、エイ アメリカ合衆国 27511 ノースカロライナ州 キャリイ、ペンウ
チ ッド ドライブ 119㉑ 出 願 人 ウェーブトレーサー インコー アメリカ合衆国 01720 マサチューセッツ州 アクトン、グレー
ボレイテッド ト ロード 289

㉒ 代 理 人 弁理士 秋元 輝雄

㉓ 指 定 国 AT(広域特許), AU, BE(広域特許), CA, CH(広域特許), DE(広域特許), DK(広域特許), ES(広域
特許), FR(広域特許), GB(広域特許), GR(広域特許), IT(広域特許), JP, LU(広域特許), NL(広
域特許), SE(広域特許)

最終頁に続く

請求の範囲

1. 選択された1つの問題次元に含まれるノードの数より少ない数のプロセッサセルを有するプロセッサアレイのための、プロセッサセル命令とそれに対応するプロセッサセル物理メモリアドレスを生成することを目的とした、仮想処理命令・アドレスジェネレーターを備えた、以下のものからなる並列処理システム:

前もって決められた数の物理プロセッサセルを有する1つのプロセッサアレイ;

解かれるべき1つの問題次元のサイズを構成する数の問題ノードを提供するための手段;

問題ノード数を提供するための手段と前もって決められた数の物理プロセッサセルに 대응して、この前もって決められた数の物理プロセッサセルの各々に伴うべき仮想プロセッサの数を確定するための手段;

前記の前もって決められた数の物理プロセッサセルの各々に伴うべき仮想プロセッサの数を確定するための手段に 対応して、物理プロセッサセルの各々に伴う各仮想プロセッサのベースメモリアドレスに対応する1つの物理プロセッサセルメモリアドレスを確定するための手段;

1つまたは1つ以上のプロセッサセル命令を提供するための手段;

各オペランドを前記のプロセッサセルのアレイによって処理するための、

1つの仮想オペランドメモリアドレスを提供するための手段;

仮想オペランドメモリアドレスを提供するための前記の手段と前記の仮想プロセッサのベースメモリアドレスに 対応して、1つの対応する仮想プロセッサベースメモリアドレスで仮想オペランドメモリアドレスを加算し、処理されるべきオペランドの1つの物理プロセッサセルメモリアドレスを生成するための手段; ならびに

処理されるべきオペランドの物理アドレスを発生させるための前記の手段および1つまたは1つ以上のプロセッサセル命令を提供するための前記の手段に 対応して、プロセッサアレイの各プロセッサセルに流すべきアドレス再配置プロセッサセル命令を発生させるための対応する1つの物理プロセッサセル

メモリアドレスをもった1つのプロセッサセル命令を生じさせるための手段

2. 請求項1のシステムにおいて、プロセッサセルアレイに、多次元アレイを構成するように相互接続された複数のプロセッサセルが含まれるもの

3. 請求項2のシステムにおいて、多次元アレイに1つの3次元アレイが含まれるもの

4. 請求項1のシステムにおいて、1つの問題次元のサイズを構成する問題ノード数を提供するための手段に、前置型プロセッサが含まれるもの

5. 請求項1のシステムにおいて、各仮想プロセッサのベースメモリアドレスに対応する1つの物理プロセッサセルメモリアドレスを確定するための手段が、オペランドがそこから取られるべき原始仮想プロセッサのベースメモリアドレスと、オペランドがそこに記憶されるべき宛先仮想プロセッサのベースメモリアドレスを確定するもの

6. 請求項1のシステムにおいて、プロセッサセル命令の各々に、少なくとも1つのプロセッサセルアドレスセグメントと1つのプロセッサセルコントロールセグメントが含まれるもの

7. 請求項1のシステムにおいて、対応する1つの物理プロセッサセルメモリアドレスをもった1つのプロセッサセル命令を生じさせるための手段が、その物理プロセッサセルメモリアドレスを、1つの対応するプロセッサセル命令のアドレスセグメントの値に加算するもの

8. 選択された1つの問題次元に含まれるノードの数より少ない数のプロセッサセルを有する多次元プロセッサアレイのための、プロセッサセル命令とそれに対応するプロセッサセル物理メモリアドレスを生成することを目的とした、仮想処理命令・アドレスジェネレーターを備えた、以下のものからなる多次元並列処理システム:

少なくとも1つの3次元アレイを形成するように相互接続された、前もって決められた数の物理プロセッサセルを有する1つのプロセッサアレイ;

解かれるべき1つの問題次元のサイズを構成する数の問題ノードを提供するための手段;

問題ノード数を提供するための手段と前もって決められた数の物理プロセッサ

サセルに匹着して、この前もって決められた数の物理プロセッサセルの各々に伴うべき仮想プロセッサセルの数を確定するための手段：

前記の前もって決められた数の物理プロセッサセルの各々に伴うべき仮想プロセッサセルの数を確定するための手段に匹着して、物理プロセッサセルの各々に伴う各仮想プロセッサセルのベースメモリアドレスに対応する1つの物理プロセッサセルメモリアドレスを確定するための手段：

プロセッサセル命令の各々に少なくとも1つのプロセッサセルアドレスセグメントと1つのプロセッサセルコントロールセグメントが含まれるような状態で、1つまたは1つ以上のプロセッサセル命令を提供するための手段：

各オペランドを前記のプロセッサセルのレイによって処理するための、1つの仮想オペランドメモリアドレスを提供するための手段：

仮想オペランドメモリアドレスを提供するための前記の手段と前記の仮想プロセッサセルのベースメモリアドレスに匹着して、1つの対応する仮想プロセッサセルベースメモリアドレスで仮想オペランドメモリアドレスを加算し、処理されるべきオペランドの1つの物理プロセッサセルメモリアドレスを生成するための手段： ならびに

処理されるべきオペランドの物理アドレスを発生させるための前記の手段および1つまたは1つ以上のプロセッサセル命令を提供するための前記の手段に匹着して、多次元プロセッサレイの各プロセッサセルに流すべきアドレス再配置プロセッサセル命令を発生させるための対応する1つの物理プロセッサセルメモリアドレスをもった前記のプロセッサセルアドレスセグメントを加算するための手段

いられる。この特許では、3次元プロセッサレイの挙動に似せてプロセッサセル間でデータの切り換えを行うよう、ルーターに命令を与えることができるようになってい。しかしながらルーターは、データを各種プロセッサの間で移動させるために、多量のプロセッサオーバーヘッドを必要とする。またこれに加えて、スイッチング機能を実行するための支援制御回路も必要である。システムスループットまたは帯幅は、ルーターオーバーヘッドによってかなり損なわれ、またシステムのコストと信頼性も、ルーターと支援制御回路を含めなければならないために、やはり大きく損なわれる。

さらにまた、並列処理問題の多くは、プロセッサレイの利用可能なプロセッサセル数を上回る数の問題ノードを必要とする。こうした状況においては、ある与えられたプロセッサレイに問題空間を合致させるための調整を、応用プログラマーに行わせることは回避するのが望ましい。むしろ応用プログラマーは、プロセッサレイのサイズの束縛から切り離し、問題空間のサイズのみを指定すればよいという状況にすべきである。さらにまた、プロセッサレイのサイズが異なる各種のプロセッサシステム間での共用性を確保するために、プロセッサレイがどのようなサイズのものであっても、並列処理プログラムをランさせることが可能でなければならない。

発明の要約：

本発明は、問題空間のノード数より少ない数のプロセッサセルを有していても差し支えない多次元プロセッサレイなどの並列処理レイに、プロセッサセル命令を生成するための、仮想処理アドレス・命令ジェネレーターを含む並列処理システムを提供するものである。

好ましい1つの実施例においては、上記処理システムは各プロセッサセルのメモリーをいくつかの等しい部分に分割し、そしてこの分割された夫々に1つの問題空間を対応させる。つぎに命令ジェネレーターは、与えられた問題空間の各部分空間に対して1続きのプロセッサセル命令を生じさせ、そしてこの命令シーケンスのそれぞれについて、該当するアドレス変更がアドレスジェネレーターによって与えられる。このような仕組みにより、該並列処理レイは、プロセッサレイに存在するプロセッサセル数より多い数の問題ノードを有する問

名称：

並列プロセッサレイのための仮想処理アドレス・命令ジェネレーター

発明の分野：

本発明はコンピューターシステム、そしてより詳細には並列処理コンピューターシステムに関するものである。

発明の背景：

電場と磁場、流体の流れ、音波、熱の流れなどの自然現象の、ある与えられた時刻における状態は、3次元の、空間的に分布するデータとして、互いに直交する3つの座標軸の方向における空間的位置が指標をもって示されるところの、1つまたは1組の数によって表現される。場合によっては、さらに第4の次元として、「時間」も考慮しなければならない。これまで科学者やその他のコンピューター・ユーザーは、ボッソソ方程式やマックスウエル方程式など、空間的に分布するデータにかかわる偏微分方程式を解こうとする場合に、十分なデータ処理能力を享受することができなかった。これは、従来技術によるプロセッサシステムとプロセッサレイが、2次元構成に限定されていたことによるものである。

純粋に1次元レイまたは2次元レイを用いて行う、3次元方程式を解くための従来技術による試みは、プロセッサではなくメモリーに記憶させなければならない空間データの、第2または第3の次元への指標またはアドレスポインターを計算するために、大量の「プロセッサオーバーヘッド」を必要とする処理システムを生み出した。たとえば伝統的なシングルプロセッサ型コンピューターの場合、X次元とZ次元の指標を計算するには、最大で3回の乗算と2回の加算を実行しなければならない。2次元コンピューターを使った場合には、X軸とY軸は直接的にアクセスできるが、Zポインターはやはり計算しなければならない。そしてこの計算のために依然として最大で1回の乗算と加算が必要である。

3次元データを従来技術によって取り扱おうとする場合には、このほか、米国特許No. 4, 814, 973で開示されているような相互結合配列もまた用

題を、メモリースワッピング無しで解くことができる。

図面の説明：

上記ならびに上記以外の本発明の特徴は、以下に述べる説明を添付図面を参照しつつ読み通すことにより、一層よく理解されるであろう。添付図面において、

図1は、本発明にもとづく、コントロールプロセッサと仮想処理アドレス・命令ジェネレーターを有する多次元並列処理システムとプロセッサレイの概要を示す。

図2は、本発明にもとづくアドレス・命令ジェネレーターの一層詳細なブロック図である。

図3は、図1に示すコントロールプロセッサの一層詳細なブロック図である。

図4は、図2に示すオペランドアドレスジェネレーターの一層詳細なブロック図である。

図5は、本発明にもとづくマイクロコードメモリアドレスジェネレーターと即時オペランドビットエクストラクターの一層詳細なブロック図である。

図6は、本発明にもとづく仮想処理アドレス・命令ジェネレーターのタイミング図である。

図7Aと図7Bは、物理・仮想プロセッサセルメモリーの分割を示す。

図8は、図2に示すアドレス再配置回路の一部分をなすアドレス再配置コントロールレジスターのブロック図である。

図9と図10は、図2に示すアドレス再配置回路の追加回路にかかわる略略図である。

図11Aと図11Bは、プロセッサ間通信命令の1例を示す。 また、

図12は、図2に示すナノ命令・テンプレートジェネレーター回路の一層詳細なブロック図である。

発明の詳細な説明：

図1は、複数のプロセッサセルを有する、本発明にもとづく多次元プロセッサシステム100を示す。該プロセッサシステムには、小型コンピューターシステムインターフェース(SCSI)106などのインターフェースによ

てプロセッサコントローラ104に接続されてコマンドプリプロセッサとして機能するところの、適切な前置型プロセッサ102が含まれる。好ましい実施例においては、この前置型プロセッサ102として、Unix™オペレーティングシステムの制御のもとに動作する通常のコンピュータを用いる。かかる前置型プロセッサの一例は、Sun Microsystems社から入手することが可能なSunワークステーションである。前置型プロセッサ102には、通常のコンパイラと、C言語への並列処理拡大体を用いて書き込みがなされたプログラムをシステムがコンパイルし実行することを可能にする、前記Cコンパイラのためのプリプロセッサが含まれる。

プロセッサコントローラ104は、前置型プロセッサ102とプロセッサアレイ108の間の中間処理機能を提供する。このプロセッサコントローラ104には、コントロールプロセッサプログラムメモリ112に記憶されたコントロールプロセッサプログラムの制御のもとに動作するコントロールプロセッサ110が含まれる。このような並列処理システムの一例は、いずれも本明細書で引用しているところの、同時懸架米国特許No. 07/530,027 (名称"MULTI-DIMENSIONAL PROCESSOR SYSTEM AND PROCESSOR ARRAY WITH MASSIVELY PARALLEL INPUT/OUTPUT") および同No. 07/529,962 (名称"DIMENSIONALLY RECONFIGURABLE MULTI-DIMENSIONAL PROCESSOR ARRAY") に開示されているとおりである。

前置型プロセッサ102はコントロールプロセッサ110と連絡し、選択されたコマンドに responding に該コントロールプロセッサにコマンドを送り、かつまた該コントロールプロセッサからのデータを読み取る。通常、各々のコマンドは、乗数や乗数などのワード・オリエンテッド型の単項演算または2項演算を記述する。コントロールプロセッサ110はかかるコマンドを固定し、そしてそれらをマクロ命令と呼ばれる1続きのワード・オリエンテッド型のコマンドとしてリフォーマットする。マクロ命令はアドレス・命令ジェネレーターに転送される。そしてこのアドレス・命令ジェネレーターは、シリアライザー内に配置

されたレジスターを初期設定することにより、シリアライザー120と命名したもことになる。シリアライザー120はマイクロロード122の制御のもとに動作し、そして各々のワード・オリエンテッド型マクロ命令を、ナノ命令と呼ばれる1続きのビット・オリエンテッド型コマンドに変換する。このビット・オリエンテッド型コマンドは、それぞれ、メモリアドレスビットとコントロールビットからなるプロセッサセル命令である。該シリアライザーは、該シリアライザーが生成させるナノ命令の各シーケンスを、プロセッサアレイ108の全てのプロセッサに伝送する。ナノ命令メモリ124には、シリアライザー120がそれにアクセスしてプロセッサアレイのためのナノ命令を形成するところの、ナノ命令テンプレートの表が含まれる。

コントロールプロセッサ110はシリアライザー120との間で直接的にデータの受渡しを行うが、一方、両方向エッジFIFO126を経由してプロセッサアレイ108の間でもデータの受渡しを行う。同様に、前置型プロセッサ102との間の通信は小型コンピュータシステムインターフェース(SCSI)106を介してなされる。コントロールプロセッサ110はプロセッサアレイ108を初期設定するためのコマンドを受け取り、そしてプロセッサアレイ、前置型プロセッサ、ならびに、標準およびオプションのスワッピングRAMやオプションのI/Oデバイスなどの周辺装置間の入出力転送を制御する。

シリアライザー120は、図2のブロック・ダイアグラムに一段詳細に示すところである。図においてシリアライザーはその入力信号をコントロールプロセッサ110から受け取り、また、コントロールプロセッサ110はその入力信号を前置型プロセッサ102から受け取る。

前置型プロセッサ102は、問題空間次元150ならびに追加的な初期設定情報をコントロールプロセッサ110へ伝送することにより、システムを初期設定する。コントロールプロセッサ110は、シリアライザー120内にレジスターをロードして、各プロセッサセルにともなう問題ノードの写像156をシリアライザーに記述することにより、この情報に responding する。アドレス再配置回路162は写像156を受け取り、そして、前置型プロセッサ102からの

後続の情報を処理する間に、それを用いて、仮想アドレスベース164と遠隔アドレスベース165の値を生成させる。

つぎに前置型プロセッサ102は、コントロールプロセッサ110にコマンドを送ってシステムを制御し始める。このコマンドは操作符号154とオペランドデータ152からなる。オペランドビット・アドレスジェネレーター168は、オペランドアドレスデータ158をコントロールプロセッサ110から受け取り、そしてプロセッサアレイによって処理されるべきオペランドビットの仮想アドレスを、信号経路170を通じてナノ命令ジェネレーター166に提供する。ナノ命令ジェネレーター166の集計回路172は、仮想アドレス170の各々を、各プロセッサセルにおいて現在のオペランドビットについて逡巡されつつある操作に応じて、仮想アドレスベース164または遠隔アドレスベース165のいずれかで加算し、処理されるべきオペランドビットのプロセッサセルメモリ物理アドレス174を形成する。

コントロールプロセッサはまた、操作符号154とオペランドデータ152を復号してマイクロプログラム制御信号160を生成させ、そしてそれをマイクロプログラム・コントロールユニット176へ送る。マイクロプログラム・コントロールユニット176は、これらの信号に responding にナノ命令テーブルアドレス178を発生し、そしてそれを用いてナノ命令メモリ124をアドレスし、ナノ命令テンプレート180を得、ついでそれをナノ命令ジェネレーター166へ送る。マイクロプログラム・コントロールユニット176はまた、オペランドビットのアドレスジェネレーター168、アドレス再配置回路162、およびナノ命令ジェネレーター166を、それ自身の操作と同期させるための制御情報を生成する。ナノ命令ジェネレーター166は、仮想アドレスベース164または遠隔アドレスベース165と仮想アドレス170の和としてそれが形成する各々の物理アドレス174を、対応するナノ命令テンプレート180のアドレスビットと加算し、それぞれのナノ命令182を生成する。それぞれのナノ命令182は、各プロセッサセルのメモリにおける1つのビットの物理アドレスと、各プロセッサセルのそのビットについて逡巡されるべき操作を記述する制御ビット、の両方を含む。ナノ命令ジェネレーター166は、それが生成させる各々の

ナノ命令182を、アレイの全てのプロセッサセルへ流す。

前置型プロセッサ102は、ワード・オリエンテッド型コマンドをコントロールプロセッサへ送ることによって、コントロールプロセッサ110と連絡する。コントロールプロセッサ110はかかるコマンドを固定し、そしてそれらを1続きのマクロ命令としてリフォーマットする。かかるコントロールプロセッサ110の1例は、図3のブロック・ダイアグラムに一段詳細に示すところであるが、マイクロプロセッサ180がこれに含まれる。このマイクロプロセッサ180はADM-29000プロセッサであって、コントロールプロセッサプログラムROMとRAM112の制御のもとに動作する。コントロールプロセッサのデータRAM114は、一時的データとプロセッサ変数記憶をマイクロプロセッサ180に提供する。

ある与えられたユーザー・プログラムを、変更せずに各種サイズのプロセッサアレイのランに使えるようにするためには、応用プログラマーは、仮想処理のためのメモリの割り付けを要求されるはならない。それゆえ、実際、応用プログラムは、ある与えられたプロセッサアレイにおける物理プロセッサセル数の束縛から切り離される。そのかわりに、応用プログラムは単に、希望する問題空間のX、YおよびZ次元のみを確定する。つぎにコントロールプロセッサは、プロセッサアレイのサイズを測定し、そして問題空間の与えられた次元とプロセッサアレイの各エッジに沿ってのプロセッサの数から、各物理プロセッサに割り振られるべき1組の仮想プロセッサを計算する。この1組の仮想プロセッサを、物理プロセッサセルの領域と称する。これらの領域次元は、好ましい実施例におけるように多次元であっても差し支えなく、次によって得られる。

$$X_v = X / s_x, \quad Y_v = Y / s_y, \quad \text{および} \quad Z_v = Z / s_z$$

ここに X_v 、 Y_v および Z_v は各物理プロセッサセルの領域の次元、 X 、 Y および Z は問題空間の次元、また s_x 、 s_y および s_z は物理プロセッサにおけるアレイの次元である。

シリアライザーは、それがコントロールプロセッサから入手することが可能な未処理の入力を有するときは、それぞれのナノ命令サイクルに1回だけ、プ

ロセッサアレイの各プロセッサセルに、有用なノ命令を渡すべきである。シリアライザは、プロセッサコントローラから受け取った各マクロ命令に対して1つのノ命令シーケンスを生成するだけでなく、また、必要に応じて各ノ命令についてアドレスを変更したうえで、既にそれぞれの物理プロセッサセルにマッピングされている各仮想プロセッサについてそのノ命令シーケンスを反復しなければならない。アドレス・命令ジェネレータはまた、2つの仮想プロセッサが相互に通信し合わなければならないときは、複数の物理プロセッサセルが関与する場合があることも考慮しなければならない。

したがって、コントロールプロセッサから受け取るある与えられたマクロ命令に対応する各ノ命令シーケンスを生成させるためには、本発明のアドレス・命令ジェネレータは、それが生成させる各ノ命令に含めるためのオペランドの物理アドレスを計算し、また、つぎのサイクルでどのマイクロコード命令を実行するか、したがってまた、どのノ命令を生成させるかを決定するための条件を評価しなければならない。つぎにシリアライザは、ある1つの物理プロセッサセルの領域にあるそれぞれの仮想プロセッサについて、該当する変更を施したうえで、それが生成させる各ノ命令シーケンスを反復しなければならない。

本発明のアドレス・命令ジェネレータ120は、dオペランドの場合について図4に示すような、5つのオペランドアドレスジェネレータを有している。マクロ命令オペランドd、eおよびfに対しては3つのアドレスユニットがあり、一方、一時的変数tおよびuに対しては2つのアドレスユニットがある。それぞれのオペランドアドレスジェネレータ168には、2つの16ビット・ハイアドレスレジスタ181と182、2つの16ビット・ローアドレスレジスタ183と184、および2つの1ビット初期設定フラグレジスタ185と186が含まれる。これらのレジスタのうちの第1のレジスタはコントロールプロセッサにより直接的にロードされる。一方、第2のレジスタはシリアライザにより第1のレジスタからロードされ、これによりシリアライザは、コントロールプロセッサが新たなマクロ命令を提供しているあいだに、1つのマクロ命令を処理することができる。このほかに、3つのカウンタすなわ

16ビット・ハイアドレスカウンタ187、16ビット位置アドレスカウンタ188、および16ビット・ローアドレスカウンタ189が含まれる。

ハイアドレスレジスタ181と182のそれぞれは、カウンタ186と同じく、ある1つのオペランドの最上位ビット(MSB)の仮想アドレスを含み、一方、ローアドレスレジスタ183と184のそれぞれは、カウンタ189と同じく、対応するオペランドの最下位ビット(LSB)の仮想アドレスを保持する。初期設定フラグ185と186は、それぞれ、当初にハイアドレスレジスタの内容がロードされるべき各ビット位置アドレスカウンタ188に対して設定され、あるいは、当初そのビット位置アドレスカウンタにロードカウンタの内容がロードされるべきであれば、クリアされる。ラン信号191はシリアライザがマイクロコードを実行しているときに設定され、そしてd-dh信号190は、dカウンタ188をdhカウンタ187からロードするマイクロ命令を実行しているときに設定される。

図4に示す5つのシリアライザアドレスユニットのレジスタとカウンタに加えて、シリアライザは、図5のレジスタ202を経由してコントロールプロセッサから情報を受け取る。マイクロプログラムアドレスレジスタ203と204、およびマイクロプログラムアドレスカウンタ206は、各シリアライザサイクルの終端でロードされる。メモリーレジスタ206はシリアライザがマイクロコードの実行を開始するときにレジスタ204からロードされ、そしてシリアライザがランしているあいだは、マイクロプログラムによって生成させられるブランチアドレスからロードされる。各シリアライザサイクルの終端でシリアライザがマイクロプログラムメモリアドレスレジスタ206をロードするときは、シリアライザはまた、アドレスレジスタ206にロードされたつづつあるアドレスで、図1のコントロールプロセッサデータメモリー114からデータワードを読み取る。このデータは、それぞれ64ビットワイドの前置オペランドレジスタF0およびF1(それぞれ210および212)に読み込まれる。fオペランドアドレスユニットからのfビットアドレスカウンタのローオーダー6ビット214は、64:1マルチプレクサ216を経由して、前置並列変数の現行ビット216となるように、F1レジスタに1

つのビットを選択する。

オペランドアドレスユニット168のタイミングは図6に示す通りである。ここにnは、各物理プロセッサセルの領域内における仮想プロセッサの数である。シリアライザはコントロールプロセッサから受け取るそれぞれのマクロ命令に対して該当するマイクロコードをn回実行し、それぞれの仮想プロセッサについて1つづつパスを行う。符号192などで示した縦線はサイクル間に選んだ境界であって、この時点でシリアライザのレジスタとカウンタにデータをロードすることができる。すなわちインクリメントが可能である。パス193などの各仮想プロセッサパスには1クロックサイクル194が先行し、そしてこのクロックサイクルの間にシリアライザは1つのデフォルト・ノ命令を発生させ、そして次のパスのためのワーキングレジスタとカウンタを初期設定する。

コントロールプロセッサは、オペランドアドレスユニットのシリアライザレジスタの第1セットdh0、d10およびdf0に、シリアライザがマクロ命令i-1(196)のための最初の仮想プロセッサパス195を開始した時点から同じくシリアライザがマクロ命令i-1のための最後のパスを終了する1クロックサイクル前の時点(197)までの間に、マクロ命令iのための該当する値をロードする。シリアライザは、それが時点198で示すマクロ命令i-1の最後の仮想プロセッサパスを終了するときに、これらの値を、バイブラインレジスタの第2のセットdh1、d11およびdf1に複写する。ついでシリアライザは、マクロ命令iのための各仮想プロセッサパスに先行する各サイクル199の終端において、ハイアドレスレジスタとローアドレスレジスタ(dh1とd11)から、3つのカウンタdh1、d、およびd1をロードする。各仮想プロセッサパスの残りのサイクルの間、シリアライザは、マクロ命令のための選択されたマイクロコードシーケンスを実行しかつマイクロコードの指令に従って各仮想プロセッサのためのアドレスカウンタを修正することにより、マクロ命令iのためのノ命令シーケンスを発生させる。

パス193などの各仮想プロセッサパスは、複数のサイクル200からなる。またシリアライザは、各サイクルのあいだに1つの有効ノ命令を発生さ

せるように設計される。

シリアライザは、各シリアライザクロックサイクル199の終端で、各オペランドアドレスユニットの3つのカウンタ187~189の内容を、インクリメントならびにディクレメントすることができる。ビット位置アドレスカウンタ188などのカウンタをインクリメントならびにディクレメントする能力は、1つのオペランドのすべてのビット位置を順々にアドレスするために必要である。コントロールプロセッサから受け取ったマクロ命令の結果として実行されるシリアライザマイクロコードは、カウンタがインクリメントされるべきかディクレメントされるべきかを指令する。

各物理プロセッサのメモリーは、図7Aのメモリーマップ220に示すように区分される。このメモリーマップには、プロセッサアレイ内の物理プロセッサ位置を記述するフラグを有する物理フラグ・セグメント222と、一時メモリー記憶領域224が含まれる。残りのメモリーは、長さがそれぞれVビットの、X、Y、Z個の等しい仮想プロセッサメモリー領域226に区分される。

それぞれの物理プロセッサセルは、各マクロ命令を、その領域にある各仮想プロセッサごとに1回づつ、全部でX、Y、Z回実行しなければならない。図7Bのメモリーマップ230は、長さVビットの、1つの仮想プロセッサメモリー226のメモリーマッピングを示す。マクロ命令間の、与えられたある1つの仮想プロセッサ226の状態は、全ての変数232の値、そのヒープ234とスタック236の状態、およびそのsフラグ238の値によって記述される。ノ命令間の仮想プロセッサの状態にはまた、マクロ命令を完成するのに必要な図7Aの一時記憶領域224の値と、物理プロセッサの各レジスタの状態が含まれる。

1つの仮想処理空間を設定するために、コントロールプロセッサは、図8の4つのプロセッサx0、y0、z0およびz0(240~246)を以下のよう

x0=X、Y、Z、V
y0=Y、Z、V
z0=Z、V

$$s0 = V$$

プロセッサアレイ内の1つのプロセッサセルから別のプロセッサセルへのデータの移動は、以下の構文に従って記述される。

$$a = [\Delta x : \Delta y : \Delta z] b$$

ただし、“a”は宛先プロセッサセルのメモリ内の1つの場所であり、また“b”は発元プロセッサセル内の1つの場所である。発元プロセッサセルは、宛先プロセッサセルaに対して座標 Δx 、 Δy 、および Δz だけ隔った場所にある。

複数の仮想プロセッサとの間でデータの移動をおこなうマクロ命令を処理するに先立ち、コントロールプロセッサは、3つのレジスタ $d \times 0$ 、 $d \times 0$ および $d \times 0$ (248~252)を以下のようにロードする。

$$d \times 0 = (\Delta x - X, |\Delta x / X, 1) Y, Z, V$$

$$d \times 0 = (\Delta y - Y, |\Delta y / Y, 1) Z, V$$

$$d \times 0 = (\Delta z - Z, |\Delta z / Z, 1) V$$

ここに Δx 、 Δy 、および Δz は、演算 $a = [\Delta x : \Delta y : \Delta z] b$ で示されるような、宛先仮想プロセッサを基準にとったときの、発元仮想プロセッサに対する符号付き仮想プロセッサ座標である。演算 $d \times 0$ 、 $d \times 0$ 、および $d \times 0$ は、距離 $d \times$ 、 $d \times$ 、および $d \times$ を該当する領域次元 X 、 Y 、および Z 、でそれぞれ除したときの剰余をあらわす。

図2のアドレス再配置回路162には再配置ベースジェネレーション回路が含まれ、そしてこの再配置ベースジェネレーション回路は仮想ベースジェネレーション回路と遠隔ベースジェネレーション回路からなる。図9の仮想ベースジェネレーション回路260には3つのモジュロコンバーター262、264および266が含まれる。第1のモジュロコンバーター262へのインクリメント入力268は、仮想プロセッサのメモリサイズ V を含むように前に定義した $s0$ レジスタの値である。各モジュロコンバーターは、そのインクリメント入力およびリミット入力各物理プロセッサセルの領域次元 X 、 Y 、および Z 、の関数として設定されるように、ステージされる。結果として得られる出力 X 、 Y および Z (270~274)は、アッダー276によって加算され仮想ベ

$$l h 0 = \begin{cases} (-1 - \Delta y) / Y, & \Delta y < 0 \\ \Delta y / Y, & \Delta y \geq 0 \end{cases}$$

$$u h 0 = \begin{cases} (-1 - \Delta z) / Z, & \Delta z < 0 \\ \Delta z / Z, & \Delta z \geq 0 \end{cases}$$

それぞれのオペランドアドレスレジスタの各々のハイアドレスカウンタは、その座標のモジュロアッダーがオーバーフローしないときはいつでも、対応する座標の物理アレイを通してデータを移動させなければならない距離を記述するように設定する。関係する座標のためのモジュロアッダーがオーバーフローしないときはいつでも、データは、もう1つの物理プロセッサを通して移動する。データを正座標の方向に移動させる場合には、それぞれのアドレスユニットのビット位置アドレスカウンタは、ハイアドレスカウンタに初期設定し、そしてローアドレスカウンタにカウントダウンする。一方、データを負座標の方向に移動させる場合には、ビットアドレスカウンタは、ローアドレスカウンタに初期設定し、そしてハイアドレスカウンタにカウントアップする。出力信号 $x + d \times$ 、 $y + d \times$ 、および $z + d \times$ (288~292)は、アッダー294によって加算され、遠隔ベース信号206を生ずる。

各仮想処理バスの開始点において、図2のマクロプログラム・コントロールユニット178は、オーバーフロービット $x0$ 、 $y0$ および $z0$ を、それぞれ、 F 、 T および U アドレスユニットのハイビット位置アドレスカウンタ187 (図4)に加算する。さらにまた、データを正の方向に移動させるべき各座標について、対応するオーバーフロービット $x0$ 、 $y0$ および $z0$ は図4の対応する位置アドレスカウンタ188にも加算される。このようにして、 F 、 T および U アドレスユニットの各々は、それぞれの仮想処理バスのあいだの物理プロセッサセルの移動数をカウントするように、正しく初期設定される。

例として、領域次元4、1、1を有する物理プロセッサ300~306のための、 $a = [-11, 0, 0] b$ の形の、プロセッサ間通信演算を図11Aおよび図11Bに示す。仮想処理を開始させた結果として、コントロールプロセッサは、各物理プロセッサの領域次元を設定するため、以下に示すように s

ス信号278が得られ、そしてそのあと、この仮想ベース信号は仮想アドレスに加算されて、アレイの全てのプロセッサセルに物理アドレスを生成させる。

図10の遠隔ベースジェネレーション回路280には3つのモジュロアッダー282、284および286が含まれ、そしてこれらが1つずつ、座標 X 、 Y および Z に対応する。これらのモジュロアッダーはそれぞれ、総和出力とオーバーフロー出力を生成させる。総和出力は次のいずれかに等しい。(a) ベース+インクリメント<リミットならば、ベース+インクリメント (b) ベース+インクリメント<リミットならば、ベース+インクリメント-リミット。オーバーフロー信号は、ベース+インクリメント<リミットならば正規出力に等しく、ベース+インクリメント<リミットならばこの入力信号の補数に等しい。

インタープロセッサ通信マクロ命令を実現するマイクロコードは、データを X 、 Y および Z 方向に移動させる物理プロセッサの数をカウントするために、それぞれ F 、 T および U を使用する。各座標でデータを移動させなければならない物理プロセッサの数を記述するため、シリアライザは、 F 、 T および U オペランドについてオペランドアドレスユニットを以下のように初期設定する。

$$f f 0 = \begin{cases} 0, & \Delta x < 0 \\ 1, & \Delta x \geq 0 \end{cases}$$

$$t f 0 = \begin{cases} 0, & \Delta y < 0 \\ 1, & \Delta y \geq 0 \end{cases}$$

$$u f 0 = \begin{cases} 0, & \Delta z < 0 \\ 1, & \Delta z \geq 0 \end{cases}$$

$$f i 0 = t i 0 = u i 0 = 1$$

$$f h 0 = \begin{cases} (-1 - \Delta x) / x, & \Delta x < 0 \\ \Delta x / x, & \Delta x \geq 0 \end{cases}$$

0、 $y0$ および $z0$ をすでにロードしている。ただし、 V は各仮想プロセッサに割り付けられたメモリービットの数である。

$$\begin{aligned} x0 &= 4V \\ y0 &= V \\ z0 &= V \\ s0 &= V \end{aligned}$$

この例の場合には、 $\Delta X = -11$ 、 $\Delta y = 0$ 、および $\Delta z = 0$ である。

図11Bの表は、各仮想プロセッサにおけるこの演算のための、キーレジスタと信号の値を示す。バス1のあいだ、仮想ベースコンバーター回路の x レジスタはクリアされており、したがって0の仮想ベースを生成する。遠隔ベースジェネレーション回路のモジュロアッダー回路は、オーバーフロー信号 $x0$ を発生し、したがって f および h のレジスタに1が加算され、これにより発元物理プロセッサ(たとえば300)が宛先物理プロセッサ(たとえば306)から3プロセッサだけ離れていることが示される。遠隔ベース(これは発元オペランドを再位置する)は V である。一方、仮想ベース(これは宛先オペランドを再位置する)は0である。これは、物理プロセッサ300の領域内にある仮想プロセッサ308から、物理プロセッサ306の領域内にある仮想プロセッサ310へのデータ移動に対応する。第2の仮想処理バスのあいだに、 x レジスタは0から V へインクリメントされ、発元または遠隔ベース0、ならびに宛先または仮想ベース V をもたらす。第4のバスではオーバーフロービット $x0$ は0になり、このため f および h のレジスタは、バスの開始点でインクリメントされない。つぎにデータは、2つの物理プロセッサのみを通して、物理プロセッサ302から物理プロセッサ306へ、そしてまた物理プロセッサ300から物理プロセッサ304へ移動する。

図2のナノ命令ジェネレーション回路166の詳細を図12に示す。この回路は、5つのビット位置カウンタ320~328、仮想ベース278、遠隔ベース296、およびナノ命令メモリ124に記憶されたナノ命令の表の数値から、ナノ命令を発生させる。マイクロプログラムの範囲のもとで、マクロ命令レジスタ330にナノ命令インデックスビット332がロードされ、そしてこれ

らのナノ命令インデックスビットが、ナノ命令メモリ124内の該当するナノ命令テンプレートを指定する1つのナノ命令アドレス334を発生させる。マルチプレクサー336は、信号338の制御のもとに、0、基底ベース296または仮想ベース278がアッダー342によって選択されたビットオペランドアドレス信号340に加算されるべきか否かを指令する。この結果としてアッダー342から得られる出力信号344は、オペランドデータの物理アドレスである。この物理アドレスはアッダー346によってナノ命令に加算され、ナノ命令レジスター348に記憶され、そして次のシリアライザーサイクルの開始時にプロセッサアレイの各プロセッサセルに渡される。

通常の技術的な熟達をもってなしうる本発明に対する変更および置換は、本発明ならびに以下添付する請求項の範囲内のものであると考える。

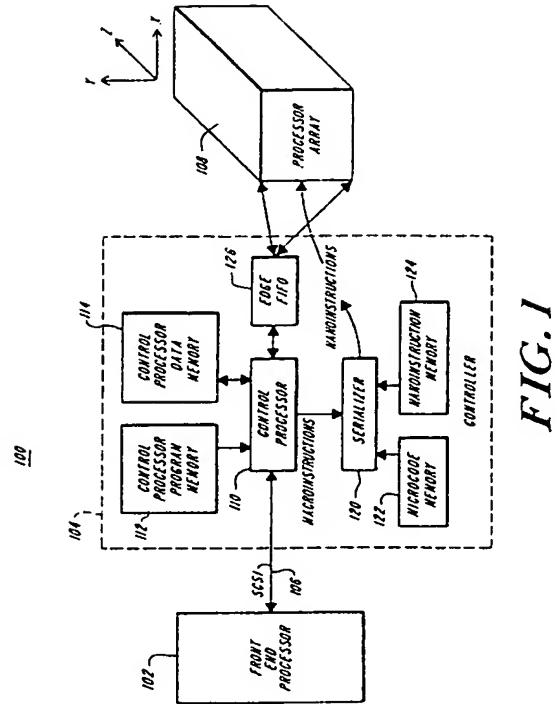


FIG. 1

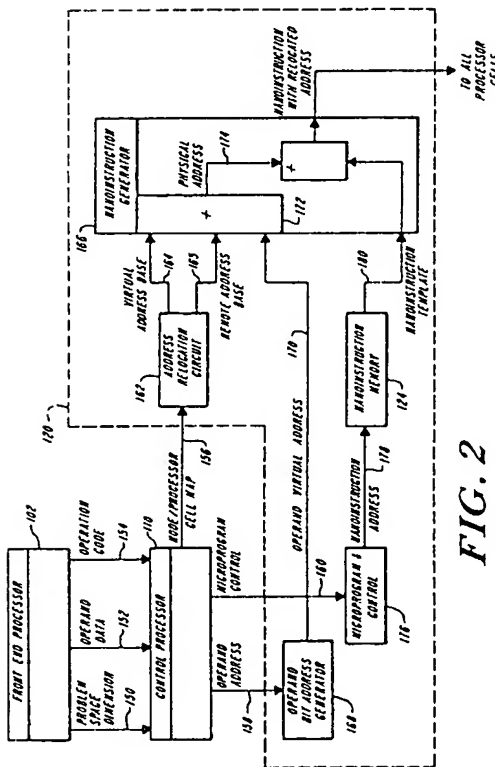


FIG. 2

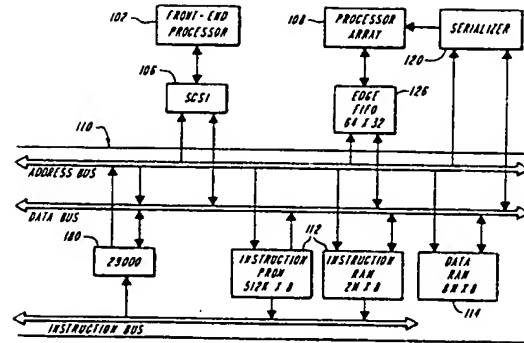


FIG. 3

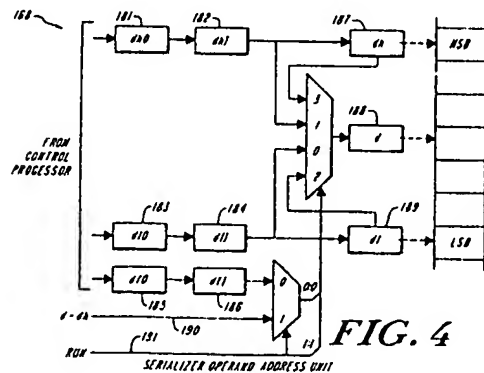


FIG. 4

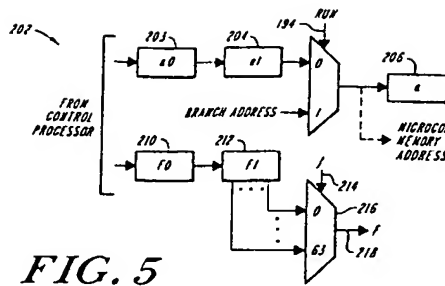


FIG. 5

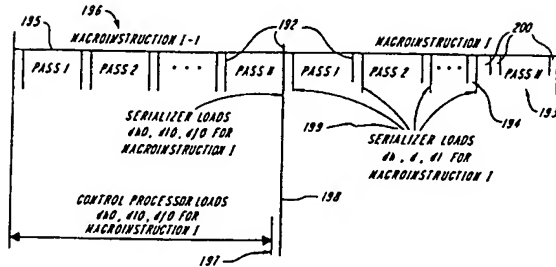


FIG. 6

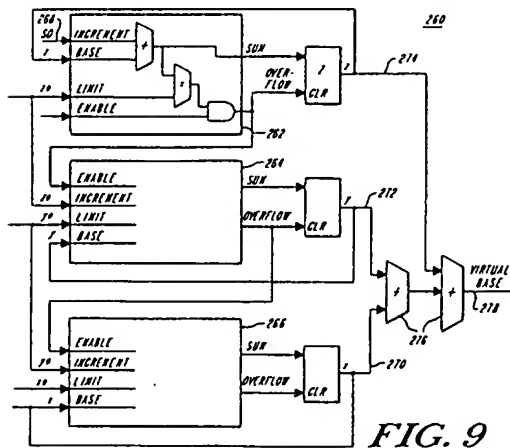
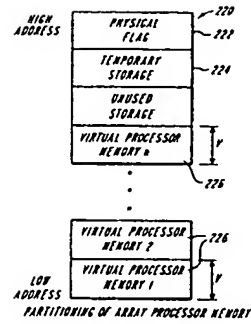


FIG. 9



第1頁の続き

②発明者 リー、ミンーチイ

アメリカ合衆国 27511 ノースカロライナ州 キヤリイ、カレン
コート 126